UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/020,033	12/06/2001	Attila D. Banki	PM 2000.063	8954

34477 7590 01/24/2008

Exxon Mobil Upstream
Research Company
P.O. Box 2189
(CORP-URC-SW 359)
Houston, TX 77252-2189

EXAMINER	
PROCTOR, JASON SCOTT	
ART UNIT	**PAPER NUMBER**
2123	

MAIL DATE	DELIVERY MODE
01/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/020,033
Filing Date: December 06, 2001
Appellant(s): BANKI ET AL.

Gary D. Lawson (27,696)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 2 November 2007 appealing from the Office action mailed 13 April 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The Math Works, Inc.; "Real-Time Workshop"; 1999; The Math Works, Inc.; Version 3, pages i through I-6

Bjarne Stroustrup, "The C++ Programming Language", 1997, AT&T Labs, Third Edition, pages 723-763

6052520 Watts 4-2000

Evidence introduced in order to respond to Appellants' arguments

The Math Works, Inc.; "SIMULINK"; 1999; The MathWorks, Inc.; Version 3

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

The Final Office Action refers to the "Real-Time Workshop" reference as "Simulink," while the Appeal Brief refers to the same reference as "Real-Time Workshop". For clarity and agreement with the Appeal Brief, the rejections shown below have been edited to refer to "Real-Time Workshop" instead of "Simulink," but the reference is unchanged.

Claims 1-13, 15-18, 20, 23-28, 30, 43, and 46 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "Real-Time Workshop" by Simulink in view of "The C++ Programming Language" by Bjarne Stroustrup.

Regarding claims 1 and 20, Real-Time Workshop teaches a processor and a memory coupled to the processor [*"The Real-Time Workshop® is a multiplatform product, running on Microsoft Windows 95, Windows 98, Windows NT, and UNIX systems."* (page xvi, first paragraph)];

Software in a main simulation system stored in the memory, the software configured to provide a logic interface to dynamically construct logic to customize simulation of a system through a model of the physical system [*"The Real-Time Workshop, for use with MATLAB and Simulink, produces code directly from Simulink models and automatically builds programs that can be run in a variety of environments, including real-time systems and stand-alone simulations."* (page 1-2, first paragraph)];

Convert the constructed logic into corresponding code during a simulation without intervention of the simulator user [*"...automatically builds programs..."* (page 1-2, first paragraph)];

Integrate, without intervention of the simulator user, the code with the main simulation system [*"Seamless integration with MATLAB and Simulink"* (page 1-3, bulleted list)] which comprises a simulation data model and simulation algorithms [*"...how the generated model code is executed. The Real-Time Workshop generates algorithmic code as defined by your model. You may include your own code into your model via S-functions."* (page 6-4, first paragraph);

"solver and data logging routines" (page 6-4, Figure 6-1), i.e. solver (algorithms) and data logs (data model) are present in the main simulation system, accessed via the Run-Time Interface];

Wherein the code extends the simulation data model by creating new classes that inherit from the simulation data model, and the code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system [*"An open and extensible architecture"* (page 1-3, bulleted list); *"Because Simulink is customizable, you can further simplify modeling by creating custom blocks and block libraries from continuous and discrete-time components."* (page 1-10, fourth paragraph); *"All Simulink blocks are automatically converted to code, with the exception of MATLAB function blocks and S-function blocks that invoke M-files."* (page 1-4, second paragraph); and "solver and data logging routines" are accessed via the Run-Time Interface (see above)]; and

Execute the integrated simulation system [*"...automatically builds programs that can be run in a variety of environments..."* (page 1-2, first paragraph)].

Real-Time Workshop does not expressly teach that the software or the code is object oriented.

Stroustrup teaches object oriented code (page 726, last paragraph).

Real-Time Workshop and Stroustrup are analogous art because both are drawn to the generation of code.

It would have been obvious to a person of ordinary skill in the art to combine the teachings of Stroustrup with Real-Time Workshop to result in object oriented software that produces object oriented code as expressly motivated by Stroustrup, because traditional design

methods (non-object oriented) are "less resilient to change, less amenable to tools, less suited for parallel development, and less suited for concurrent execution." (page 726, last paragraph).

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Stroustrup and Real-Time Workshop to arrive at the invention specified in claim 1.

Regarding claim 2, Real-Time Workshop teaches that the constructed logic comprises facility management logic which is representative of steps used to simulate the monitoring and controlling of mechanical facilities associated with the physical system [*"The first step in the design process is to develop a plant model. You can build models involving plant, sensor, and actuator dynamics using the Simulink collection of linear and nonlinear components."* (page 1-10, fourth paragraph); *"Rapid prototyping begins with developing block diagram plant models of sufficient fidelity for preliminary system design and simulation."* (page 1-10, third paragraph)].

Regarding claims 3 and 24, Real-Time Workshop teaches that the logic interface comprises a logic flow chart interface [*"Stateflow is a graphical design and development tool for complex control and supervisory logic problems. It supports flow diagram notation as well as state transition notation."* (page 1-21, first paragraph)].

Regarding claims 4-6 and 26, Real-Time Workshop teaches that the logic flow chart interface comprises arrows (page 12-21). Simulink teaches that the logic flow chart interface comprises icons representing basic logic control constructs for looping, decision making,

statement execution, and logic entry and exit [*"Stateflow supports flow diagram notation as well as state transition notation..."* (page xix, third paragraph and bulleted list)].

Regarding claim 7, Real-Time Workshop teaches that the logic interface comprises a text-based logic code interface [*"To display the model with Simulink, at the MATLAB command line, enter:* f14" (page 2-12, last paragraph)].

Regarding claims 8 and 27, Real-Time Workshop teaches that the text-based logic code interface comprises a graphical text editor for performing one or more of entering, modifying and deleting lines of alpha-numeric text [*"To display the model with Simulink, at the MATLAB command line, enter:* f14" (page 2-12, last paragraph); *"This is the Simulink model that appears:"* (page 2-13, Figure 2-1)].

Regarding claims 9 and 13, Real-Time Workshop teaches that the text-based logic code is a facility management control language automatically created from a logic flow chart [*"The Real-Time Workshop, for use with MATLAB and Simulink, produces code directly from Simulink models and automatically builds programs that can be run in a variety of environments, including real-time systems and stand-alone simulations."* (page 1-2, first paragraph)].

Regarding claims 10 and 12, Real-Time Workshop teaches that the facility management control language is automatically converted into C++ code (page xxiv). The combination formed in the rejection of claim 1 teaches the generation of object oriented code.

Claim 11 recites "wherein the facility management control language is an object-oriented language that is parsed prior to conversion into the object-oriented-facility management code to verify syntax" which is an implicit step in any computer compiler. This step is regarded as implicit in the system of claim 1 in order to fulfill the requirements of 35 U.S.C. § 112, first paragraph, as regards enablement. The combination formed in the rejection of claim 1 expressly discloses a computer compiler (ex. page xxiv) and a person of ordinary skill in the art of computer software is presumed to be aware that a computer compiler parses code to verify syntax.

Regarding claim 15, see Real-Time Workshop, page 1-12, "UNIX versions of make", and 2-18, "GNU's gcc compiler". Applicants' claim language describes the well-known method employed by GNU's gcc compiler.

Regarding claim 16, Real-Time Workshop, see page xxiv, "Microsoft Visual/C++". Applicants' claim language describes the well-known method employed by a Microsoft C++ compiler.

Regarding claims 17, 43, and 46, see page Real-Time Workshop, 1-19, last paragraph, "MdlOutputs must be called at every time step by the run-time interface."

Regarding claim 18, see Real-Time Workshop, page xvi, "Microsoft Windows 95". Applicants' claim language describes the well-known method of executing software employed by Windows 95.

Regarding claim 23, Real-Time Workshop teaches that construction of the logic comprises using a graphical user interface to perform at least selecting and using an existing logic [*"To display the model with Simulink, at the MATLAB command line, enter:* f14" (page 2-12, last paragraph); *"This is the Simulink model that appears:"* (page 2-13, Figure 2-1)].

Regarding claim 25, see Real-Time Workshop, page 1-4, second paragraph.

Regarding claim 28, see Real-Time Workshop, page xxiv.

Regarding claim 30, see Real-Time Workshop, page 1-10.

Claims 19 and 31 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "Real-Time Workshop" by Simulink in view of "The C++ Programming Language" by Bjarne Stroustrup as applied to claims 1 and 20, and further in view of Official Notice.

Regarding claims 19 and 31, Official Notice has previously been taken regarding the use of a plurality of connected processors to perform a simulation.

Claims 21-22 and 44-45 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "Real-Time Workshop" by Simulink in view of "The C++ Programming Language" by Bjarne

Stroustrup as applied to claims 1, 20, and 43, and further in view of US Patent No. 6,052,520 to Watts.

Regarding claims 21-22, and 44-45, Real-Time Workshop in view of Stroustrup does not expressly teach that the physical system comprises a hydrocarbon bearing subterranean formation or that the physical system comprises fluid-containing facilities associated with production of hydrocarbons from the hydrocarbon-bearing subterranean formation.

Watts discloses this intended use for the system of Real-Time Workshop in view of Stroustrup (abstract, entire reference).

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Watts with Real-Time Workshop in view of Stroustrup as expressly motivated by Watts, in order to understand the complex chemical, physical, and fluid flow processes occurring in a petroleum reservoir sufficiently well to be able to predict future behavior of a reservoir and to maximize recovery of hydrocarbons (column 1, lines 16-34).

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Watts with Real-Time Workshop in view of Stroustrup to arrive at the invention specified by claims 21-22 and 44-45.

(10) Response to Argument

In order to better explain the teachings of the prior art as would be perceived by a person of ordinary skill at the time of Applicants' invention, the Examiner provides the following technical background information, which should not be interpreted as a new ground of rejection.

The Math Works, Inc.; "SIMULINK"; 1999; The MathWorks, Inc.; Version 3, cited on the PTO-892 entered on 13 April 2007, provides an explanation of the "Simulink Dynamic System Simulation for MATLAB" and is hereafter referred to as "Using Simulink".

"Using Simulink" teaches that **the "Simulink Dynamic System Simulation for MATLAB" may be used with "The Simulink Real-Time Workshop"** (page 1-10). (The "Real-Time Workshop" manual is relied upon in the rejections under appeal and will be discussed below.)

"Using Simulink" teaches that **"Simulink has become the most widely used software package in academia and industry for modeling and simulating dynamical systems."** (page 1-2) A standard computer executing Simulink is "a computer system for simulating a physical system".

"Using Simulink" teaches that **"You can easily build models from scratch, or take an existing model and add to it. Simulations are interactive, so you can change parameters 'on the fly' and immediately see what happens"** (page 1-2) Simulink therefore "provides a logic interface to dynamically construct logic to customize simulation of [a physical system] through a model of the physical system.

"Using Simulink" teaches that **"The Simulink Real-Time Workshop® automatically generates C code directly from Simulink block diagrams."** To do so, **"Simulink is required."** (page 1-10) Simulink, in conjunction with Real-Time Workshop, "converts the constructed logic into corresponding code during a simulation without intervention of the simulator user."

The Math Works, Inc.; "Real-Time Workshop"; 1999; The MathWorks, Inc.; Version 3, is relied upon in the appealed rejections under 35 U.S.C. § 103 and is referred to as "Real-Time Workshop".

"Real-Time Workshop" teaches that **"The Real-Time Workshop requires: Matlab® 5.3 (Release 11), Simulink® 3.0 (Release 11)"** (page xvi).

"Real-Time Workshop" teaches that "MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include [...] Modeling, simulation, and prototyping" (page xvii). "Real-Time Workshop" describes MATLAB as "an interactive system," whose name stands for "matrix laboratory." MATLAB features "toolboxes" which "are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems" in areas such as "simulation". (all from page xvii)

"Real-Time Workshop" teaches that "The Real-Time Workshop, for use with MATLAB and Simulink, **produces code directly from Simulink models and automatically builds**

programs that can be run in a variety of environments, including real-time systems and stand-alone simulations." (page 1-2)

The Examiner submits that a person of ordinary skill in the art at the time of Applicants' invention, in possession of the "Real-Time Workshop" reference, would understand that the teachings of the "Real-Time Workshop" are applicable in combination with the "Simulink" and "MATLAB" products. A person of ordinary skill in the art in possession of the "Real-Time Workshop" reference would understand each Simulink's features to be present when using the Real-Time Workshop, because Real-Time Workshop requires Simulink and MATLAB to operate. In any event, the "Using Simulink" reference was cited on the same PTO-892 and provided in its entirety to Appellants in conjunction with the "Real-Time Workshop" reference. (PTO-892, 13 April 2007)

The rejections under 35 U.S.C. § 103 are therefore based upon the "Real-Time Workshop" reference.

Response to Argument 1

On page 4 of the Appeal Brief, Appellants' submit that:

> As a preliminary matter, in the final Office Action, the Examiner referred to *Real-Time Workshop* as "Simulink", apparently believing that *Real-Time Workshop* is made by Simulink. However, *Real-Time Workshop* is owned by The Math Works Inc. and is used with Simulink, another product by The Math Works. Because Simulink and *Real-Time Workshop* are separate products, to avoid confusion, the document entitled *Real-Time Workshop* is referred to herein by that name.

The Examiner responds as follows.

In the final office action, the Examiner referred to the "Real-Time Workshop" reference as "Simulink" because, according to both the "Real-Time Workshop" and the "Using Simulink" references, the "Real-Time Workshop" product can only be used in conjunction with the "Simulink" product. However, the Examiner has adopted Appellants' terminology for this Answer.

Response to Argument 2

On page 4 of the Appeal Brief, Appellants' submit that:

> As a second preliminary matter, in the Advisory Action, the Examiner made specific assertions that the Applicants believe should be clarified. To begin, the Examiner acknowledged in both the final Office Action and the Advisory Action that the cited references do not recite the claim language verbatim. [...] That is, the Examiner appears to assert that each of the claim elements does not have to be shown in the prior art references.

The Examiner respectfully traverses this argument as follows.

Appellants' argument attempts to equivocate the Examiner's correct interpretation of MPEP 2131 (instructing that in order to anticipate the prior art, "[t]he elements must be arranged as required by the claim, but this is not an *ipsissimis verbis* test, i.e., identity of terminology is not required") with the incorrect notion that the prior art is not required to show every element of the claimed invention. The Examiner has merely adopted these instructions in the context of a rejection under 35 U.S.C. § 103 to conclude that in order to teach a particular limitation, **the prior art is not required to recite the claim language verbatim**. This statement is clearly very different than Appellants' incorrect conclusion that the Examiner asserts "that each of the claim elements does not have to be shown in the prior art references."

This argument has been fully considered but has been found unpersuasive.

Response to Argument 3

Appellants' argument summarizes the claimed invention and argues, on page 7, that:

> *Real-Time Workshop* discloses a software package for modeling, simulating and analyzing dynamic systems utilizing a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. While *Real-Time Workshop* discloses a graphical user interface (GUI) for building models that "can run in a variety of environments, including real-time and stand-alone simulations," there is nothing in *Real-Time Workshop* that suggests **constructing logic** to customize simulation of transport phenomena, nor is there any suggestion in *Real-Time Workshop* of **converting constructed logic** into object-oriented code. (emphasis in original)

The Examiner respectfully traverses these arguments as follows.

Regarding Appellants' emphasis on "constructing logic":

Appellants' argument asserts that Real-Time Workshop discloses "a software package ... utilizing a graphical user interface (GUI) for building models as block diagrams." Real-Time

Workshop teaches on page 1-3 that "You can design your **control system** using MATLAB and Simulink and generate code from your block diagram model. You can then compile and download it directly to your target hardware" (emphasis added). A person of ordinary skill in the art at the time of Applicants' invention would recognize a "control system" as comprising logic. Further, Real-Time Workshop can only be used in conjunction with Simulink and MATLAB, as has been shown above, and "Using Simulink" clearly teaches constructing logic (Using Simulink, Chapter 3, "Creating a Model"). Therefore, Appellants' conclusion that Real-Time Workshop does not teach "constructing logic" is incorrect.

Regarding Appellants' argument that Real-Time Workshop does not teach constructing logic "to customize simulation of transport phenomena":

Appellants define the term "transport phenomena" in the specification at page 7 as:

> The term "transport phenomena" is used in this description in a broad sense to include momentum transport (viscous flow), energy transport (heat conduction, convection, and radiation), and mass transport (diffusion).

Real-Time Workshop teaches simulation of transport phenomena on page 1-3 by teaching that "[y]ou can create Simulink models that mimic real-life measurement, system dynamics, and actuation signals. Generated code from the model can be targeted on special purpose hardware to provide a real-time representation of the physical system."

Further, Real-Time Workshop teaches an example model on page 2-13 of an "F14 Aircraft" including an "Aircraft Dynamics Model," "Dryden Wind Gust Models," takes as input "the pilot's stick input" and produces "the aircraft angle of attack and the G forces experienced by the pilot."

In a second interpretation, the recitation of "transport phenomena" in the claim is a statement of intended use of the claimed computer system and therefore does not further define the claimed invention. There is no other claim language related in any way to "transport phenomena". As a result, provided adequate written description existed, Appellants' might have claimed "simulation of automobiles" or "simulation of a dynamical system" without changing or limiting any of the structural characteristics of the claimed computer system. Real-Time Workshop acknowledges what one chooses to simulate is irrelevant to the system by teaching on page 2-13 "The design and internal workings of the f14 model are not discussed in this example; such a discussion is beyond the scope of this manual. However, the procedure you must follow to use the Real-Time Workshop is generally independent of model content and complexity."

Regarding Appellants' argument that Real-Time Workshop does not teach converting constructed logic into object-oriented code:

Real-Time Workshop clearly teaches converting constructed logic into code on page 1-2 by teaching that "[t]he Real-Time Workshop, for use with MATLAB and Simulink, produces

code directly from Simulink models". Real-Time Workshop has not been relied upon to teach object-oriented code.

This argument has been fully considered but has been found unpersuasive.

Response to Argument 4

On page 8 of the Appeal Brief, Appellants argue that:

> The steps disclosed in *Real-Time Workshop* (1) do not take place without intervention of the user, much less does it occur in a single run as in Applicants' computer system and (2) the resulting code is procedural (not object-oriented), and without any teaching or suggestion of integration between data models.

The Examiner respectfully traverses this argument as follows.

Applicants' claim language broadly recites "without intervention of the simulator user". Accordingly, a broad interpretation of what, precisely, is done "without intervention of the simulator user," is appropriate within the confines of the claim language.

Real-Time Workshop describes automatically building programs on page 1-2. Real-Time Workshop describes the automated process of generating code from a block model on pages 1-12 through 1-22. These pages show numerous examples of how Real-Time Workshop operates "without intervention of the simulator user," for example on page 1-17:

> When you invoke the Real-Time Workshop build procedure to generate code, it first compiles your model. This compilation process consists of these tasks: Evaluating simulation and block parameters; Propagating signal width and sample times; Computer work vector sizes such as those used by S-functions [...]; Determining the execution order of blocks within the model. The Real-Time Workshop writes this information out to example.rtw.

Secondly, Appellants' present no claim language that requires that the steps be performed "in a single run". It is the claims, not the disclosure, which must define the patentable invention.

Thirdly, Real-Time Workshop has not been relied upon to teach the generation of object-oriented code. Nevertheless, Real-Time Workshop, on page 6-4, teaches (emphasis added):

> The Real-Time Workshop also provides a run-time interface that executes **the generated model code**. **The run-time interface and model code are compiled together to create the model executable**. The diagram below shows a high-level **object-oriented view of the executable**.

Lastly, Real-Time Workshop teaches integration between data models as shown above (page 6-4) as well as on page 1-14 ("The Real-Time Workshop passes model.mk to the make utility, which directs the compilation and linking of model code [i.e. "constructed logic converted into code"] along with any libraries ["data model and algorithms of the main simulation system"] or user provided models"), and pages 1-15 through 1-22 generally describing how Real-Time Workshop combines the user's model with the necessary libraries and simulator functions to produce a "stand alone executable," which is a computer program that executes a simulation of the user's model independently and without requiring the Simulink simulation software.

This argument has been fully considered but has been found unpersuasive.

Response to Argument 5

On pages 9-11 of the Appeal Brief, Appellants compare the claim language to the cited portions of the prior art references. These pages present numerous conclusions without citations of factual basis or with rationale that the Examiner finds faulty. The following is an attempt to address these pages of the Brief completely.

Appellants argue on page 9 that:

> C is suited for procedural code, and does not naturally support object-orientation. C++ code by its nature supports object-oriented principles, however, code written in C++ could be plain procedural code, and C++ code is not necessarily object-oriented.

Stroustrup has been cited as **teaching** object-oriented code in C++, and has been shown to provide **motivation** to a person of ordinary skill in the art for producing **object-oriented** C++

code. The fact that it is possible to do something else with C++ is not relevant to the present issue of patentability.

This argument has been fully considered but has been found unpersuasive.

Appellants argue in several instances that Real-Time Workshop does not generate object-oriented code. The Examiner submits that this is piece-meal analysis of a combination of references relied upon in a rejection under 35 U.S.C. § 103. Stroustrup teaches object-oriented code.

This argument has been fully considered but has been found unpersuasive.

Appellants argue on page 10 that:

> One critical point is that the integration goes well beyond invoking code and passing data. The claimed subject matter integrates in a more profound way by extending the existing main simulation system during the simulation and without intervention of the simulator user.

The Examiner respectfully submits that Appellants have not particularly claimed what Appellants describe as a "critical point". The Examiner has applied the prior art to the expressly recited language of the claim. Real-Time Workshop teaches "extending" the existing main simulation system during the simulation and without intervention of the simulator user (page 6-4, compilation of the run-time interface and the model to form an executable, for example). Appellants appear to be arguing a difference in vocabulary where the prior art clearly teaches the structure and functionality of the claimed invention.

This argument has been fully considered but has been found unpersuasive.

Appellants argue on page 11 that:

> Although Real-Time Workshop refers to the simulation as having "extensible architecture" and the Real-Time Workshop simulation is stated as being "customizable", these phrases in no way suggest object-oriented functionality. This flexibility available to Real-Time Workshop users refers to Real-Time Workshop's ability to invoke additional components, and does not refer to the type of flexibility in Applicant's claimed subject matter, because Real-Time Workshop can not teach characteristics of an object-oriented system since it does not generate object-oriented code.

The Examiner respectfully traverses this argument as follows.

It is noted the Appellants do not provide citations for their conclusions about the capabilities of Real-Time Workshop.

The Examiner has shown where and how Real-Time Workshop teaches the relevant limitations of the claimed invention.

The Examiner submits that "object-oriented" is a method of software **development**, and is therefore germane to a discussion of **software development techniques** and does not result in patentably distinguishable software. The range software that can be produced by object-oriented

or non-object-oriented design is identical. Any program written in object-oriented design may be equivalently produced using a non-object-oriented design.

This argument has been fully considered but has been found unpersuasive.

Appellants argue on page 11 that:

The Examiner's reference to MATLAB is misplaced because MATLAB is a programming language for technical computer; MATLAB is not a simulator. The Examiner's assertion that the phrase "seamless integration with MATLAB and Real-Time Workshop" comprises a simulation data model and simulation algorithms is also misplaced because the term "data model" is defined in the present application to mean " a collection of C++ classes" (page 26, line 13), which are based upon object-oriented principles, which as discussed above, Real-Time Workshop does not support.

The Examiner respectfully traverses this argument as follows.

"Real-Time Workshop" teaches that "MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include [...] Modeling, simulation, and prototyping" (page xvii). "Real-Time Workshop" describes MATLAB as "an interactive system," whose name stands for "matrix laboratory." MATLAB features "toolboxes" which "are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems" in areas such as "simulation". (all from page xvii)

Appellants' argument appears to overlook the combination of references used in the rejection. Stroustrup clearly teaches object-oriented code. A product such as Simulink or Real-Time Workshop, when developed in object-oriented C++ code, clearly comprises "a collection of C++ classes".

This argument has been fully considered but has been found unpersuasive.

On page 12 of the Appeal Brief, Appellants argue that:

> Simply put, neither Real-Time Workshop nor Stroustrup teach or suggest how to integrate, without intervention of the simulator user, the object-oriented code with the main simulation system and doing so during a simulation without intervention of the simulator user.

The Examiner respectfully traverses this argument as follows.

The "integration" functionality is shown in Real-Time Workshop as demonstrated above, and largely described in chapter 6 of that reference. The "object-oriented code" limitations are taught by Stroustrup.

This argument has been fully considered but has been found unpersuasive.

In conclusion, the Examiner submits that the references relied upon in the rejections under 35 U.S.C. § 103 render the claimed invention obvious over the prior art for the reasons set forth above.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Jason Proctor

Conferees:

Paul Rodriguez

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Eddie Lee

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER